ADVANCED PERFORMANCE MEASUREMENT WITH CISCO IOS IP SLAs

BRKNMS-3004

Emmanuel Tychon

Cisco Networkers 2007

HOUSEKEEPING

- We value your feedback, don't forget to complete your online session evaluations after each session and complete the Overall Conference Evaluation which will be available online from Friday.
- Visit the World of Solutions on Level -01!
- Please remember this is a 'No Smoking' venue!
- Please switch off your mobile phones!
- Please remember to wear your badge at all times including the Party!

Prerequisites

- Before attending this session, familiarities with Cisco IOS® IP Service Level Agreements (IP SLAs) is essential.
- Configuration and generic features will not be covered.
- Only new or advanced topics, as well as design recommendations will be covered

Objectives

- This session targets *network performance measurement* only.
- Understand the internals
- New features update
- Performance and scalability considerations
- How to get the most of IP SLAs
- Future and IP SLAs strategic vision

This Is Not…

- An introduction to IP SLAs
- Recommendations on QoS configuration
- A talk on backend network management applications
- A speculation on upcoming features
- …a Marketing document

Agenda

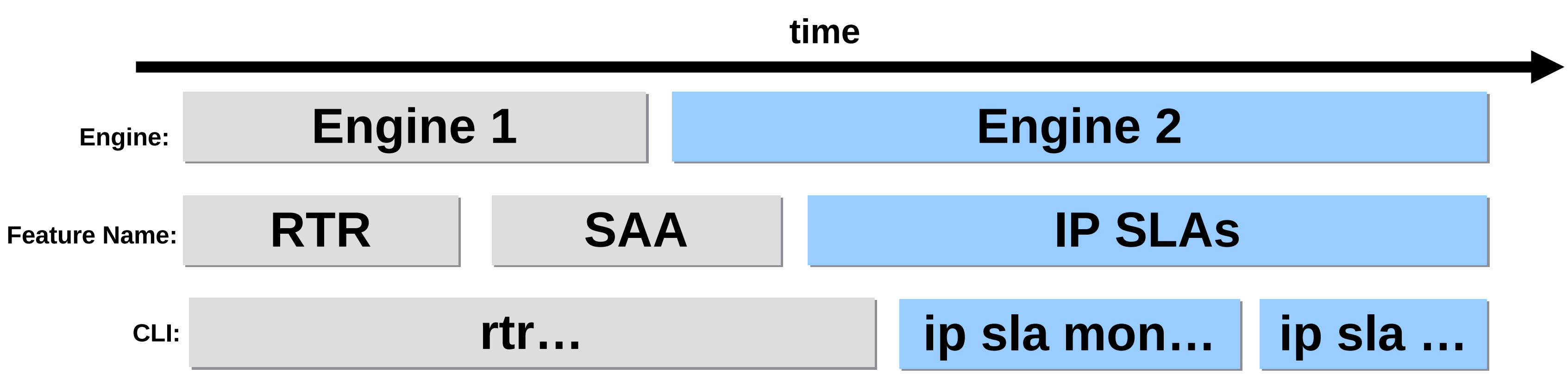
- Reminder
- IP SLAs Accuracy
- Performance and Scalability
- New Features
- Design Recommendations
- Get the Most Out of IP SLAs
- IP SLAs Initiative

Reminder

- IP SLAs in an active probing and monitoring feature in Cisco IOS
- Wide protocol and applications coverage: UDP, TCP, ICMP, HTTP, DNS, DHCP, FTP,…
- Microsecond granularity
- Use it through SNMP or CLI
- Already in Cisco IOS® (available on most platforms and interfaces type)

IPSLA History

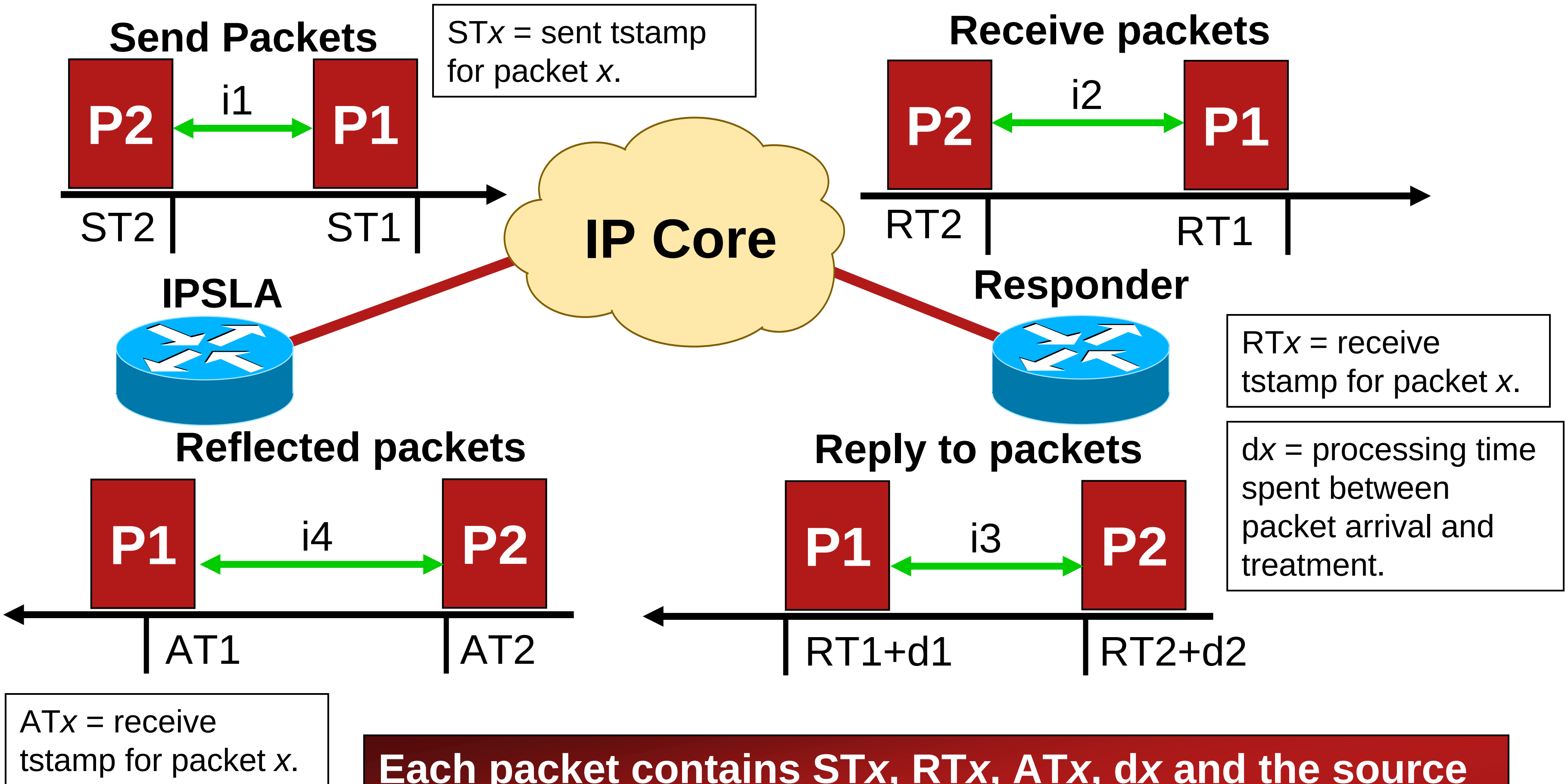
- Used to be called RTR, renamed SAA in 12.0(5)T; we call it “IP SLAs Engine 1”.
- New “IP SLAs Engine 2” is a major code rewrite to improve speed and memory usage. Introduced initially in 12.2(15)T2, 12.3(3) and 12.2(25)S, and is therefore present in all later trains. Also planned for 12.0(32)SY and 12.2(18)SXG.
- First phase of new CLI appears originally in 12.3(14)T, next phase for 12.4(6)T. MIBs are unchanged.



UDP Jitter Operation

- Measures the delay, delay variation (jitter), corruption, misordering and packet loss by generating periodic UDP traffic
- One-way results for jitter and packet-loss. If clocks are synchronized and IOS is at least 12.2(T), one-way delay is also measured.
- Detect and report out-of-sequence and corrupted packets
- Since 12.3(4)T -- also with MOS and ICPIF score for voice clarity estimation.
- This operation always requires IPSLA responder

UDP Jitter - Measurement Example

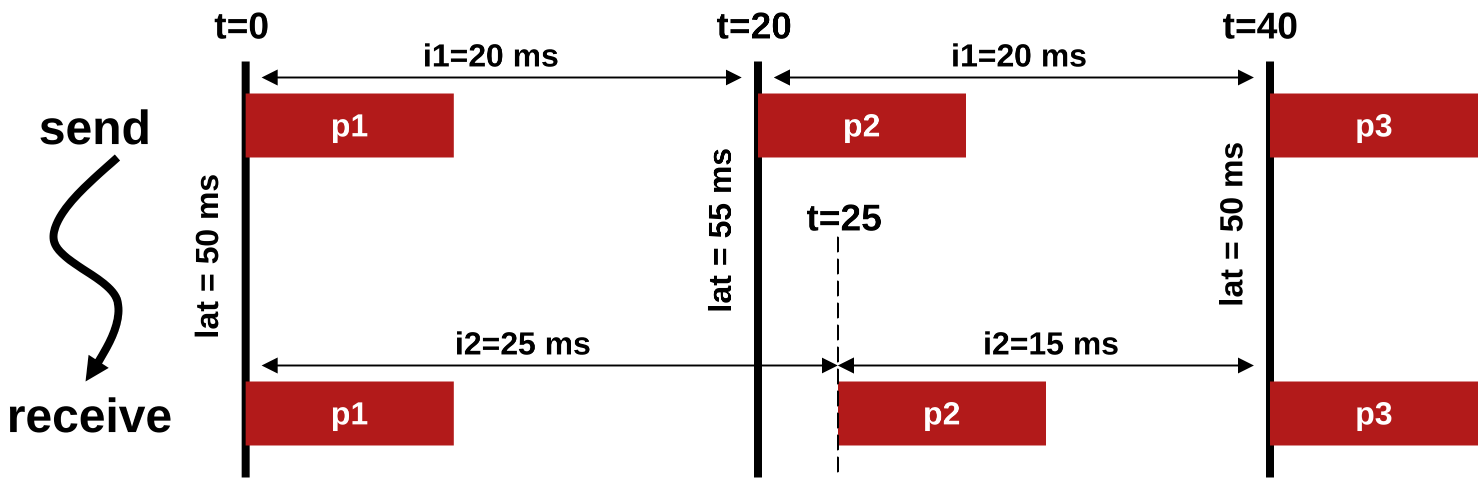


Each packet contains STx, RTx, ATx, dx and the source can now calculate:

JitterSD = $(RT2-RT1)-(ST2-ST1) = i2-i1$

JitterDS = $(AT2-AT1)-((RT2+d2)-(RT1+d1)) = i4-i3$

Jitter Calculation – beware!



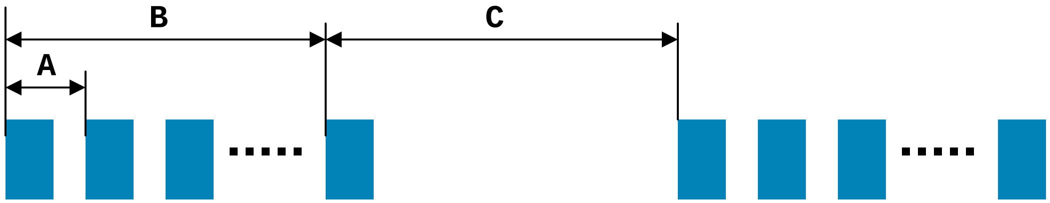
IPSLA Jitter	i2-i1 = +5 ms	i2-i1 = -5 ms
RTP Stream Jitter	Packet too late: 5 ms	Packet on-time: 0 ms

If you count positive AND negative jitter, you are penalized twice. Counting only positive jitter is enough.

UDP Jitter Operation (Example)

- Simulating G.711 VoIP call
- Use RTP/UDP ports 16384 and above, the packet size is 172 bytes (160 bytes of payload + 12 bytes for RTP)
- Packets are sent every 20 milliseconds
- Marked with DSCP value of 8 (TOS equivalent 0x20)

```
ip sla 1
  udp-jitter 10.52.130.68 16384 \
       num-packets 1000 interval 20
 tos 0x20
 frequency 60
 request-data-size 172
ip sla schedule 1 life forever start-time now
```



A = 20 ms
B = 20 s (1000 x 20 ms)
C = 40 s (60 s – 20 s)

UDP Jitter Example (new CLI Phase I)

- Differences between CLIs:

```
rtr 1
  type jitter dest-ipaddr 10.52.130.68 dest-port 16384 \
          num-packets 1000 interval 20
  request-data-size 172
  tos 20
  frequency 60
rtr schedule 1 life forever start-time now
```

```
ip sla monitor 1
  type jitter dest-ipaddr 10.52.130.68 dest-port 16384 \
       num-packets 1000 interval 20
  request-data-size 172
  tos 20
  frequency 60
ip sla monitor schedule 1 start-time now
```

```
ip sla 1
  udp-jitter 10.52.130.68 16384 \
          num-packets 1000 interval 20
  request-data-size 172
  tos 20
  frequency 60
ip sla schedule 1 life forever start-time now
```

UDP Jitter Output

```
etychon-1#sh ip sla statistics 1
Round trip time (RTT)   Index 1
        Latest RTT: 1 ms
Latest operation start time: *10:33:11.335 PST Fri Oct 7 2005
Latest operation return code: OK
RTT Values
        Number Of RTT: 20
        RTT Min/Avg/Max: 1/1/4 ms
Latency one-way time milliseconds
        Number of Latency one-way Samples: 10
        Source to Destination Latency one way Min/Avg/Max: 0/0/0 ms
        Destination to Source Latency one way Min/Avg/Max: 1/2/4 ms
Jitter time milliseconds
        Number of Jitter Samples: 19
        Source to Destination Jitter Min/Avg/Max: 4/4/4 ms
        Destination to Source Jitter Min/Avg/Max: 3/3/3 ms
Packet Loss Values
        Loss Source to Destination: 0           Loss Destination to Source: 0
        Out Of Sequence: 0      Tail Drop: 0    Packet Late Arrival: 0
Voice Score Values
        Calculated Planning Impairment Factor (ICPIF): 0
        Mean Opinion Score (MOS): 0
Number of successes: 5
Number of failures: 3
Operation time to live: 3166 sec
```

UDP Jitter with VoIP MOS Score

- Newly introduced in Cisco IOS 12.3(4)T—IP VOICE feature set
- Modified jitter operation reports both Mean Opinion Score (MOS) and Calculated Planning Impairment Factor (ICPIF)
- Those results are estimates and should be used for comparison only and should not be interpreted as reflecting actual customer opinions
- Supported Codecs:

  G.711 A Law (g711alaw: 64 kbps PCM compression method)

  G.711 mu Law (g711ulaw: 64 kbps PCM compression method)

  G.729A (g729a: 8 kbps CS-ACELP compression method)

VoIP Operation: Sample Configuration

- Autoconfigured to simulate a G729a codec
- Default is 1000 packets, interval 20 ms
- Operation frequency is random between 40 and 60 seconds

```
ip sla 30
 udp-jitter 192.1.3.2 16001 codec g729a
ip sla group schedule 30 30-31 schedule-period 1
frequency range 40-60 start-time now life forever
```

Availability

- Available on all platforms running Cisco IOS, with a few exceptions.
- Catalyst 4500 is not supported.
- Catalyst 3000 series is not supported, with the exception of the Catalyst 3550.

Summary

- Cisco IOS Feature
- Active monitoring with synthetic operations
- Detailed results like delay, loss, and jitter per direction and MOS score.
- Easy to use, available on many platforms.

Agenda

- Reminder
- IPSLA Accuracy
- Performance and Scalability
- New Features
- Design Recommendations
- Get the Most Out of IPSLA
- IPSLA Initiative

IPSLA Accuracy...ICMP Echo Probe

(90% Process Load)

- **With unloaded receiver, IPSLA measures 15.0 ms**
- **With high CPU load on the receiver: 58.5 ms!!**

Any System Will Report Wrong Results when Excessive CPU Time Is Spent on the Receiver Between the ICMP Echo Request and Echo Reply

Fortunately, We Have a Solution…

Processing Time Measurement

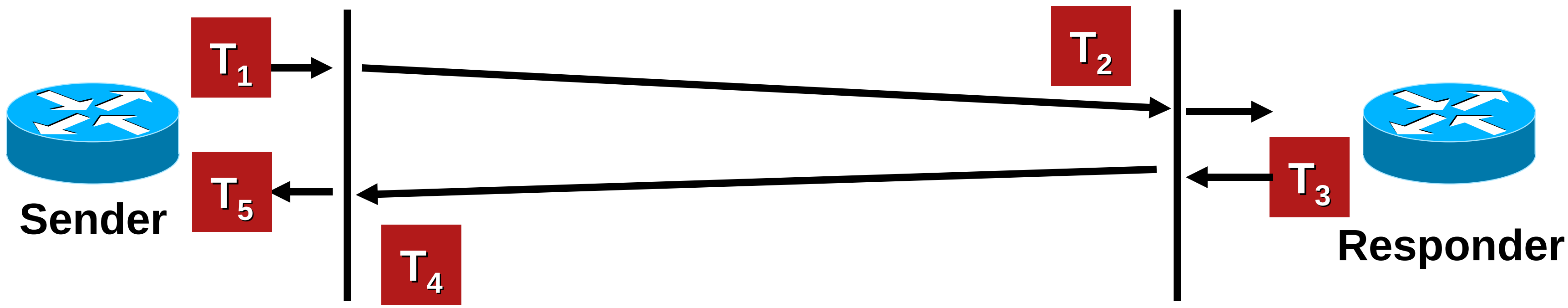
- When running the responder, we have a clear advantage, because…

  A mechanism to measure the processing time spent on the receiving router is in place, inserting a timestamp when the responder receives and send the packet

  Receive timestamp done at interrupt level, as soon as the packet is dequeued from the interface driver; with absolute priority over everything else

- With IPSLA, this mechanism is implemented for both UDP Echo and UDP Jitter operations

UDP Echo Operation (With IPSLA Responder)



Processing Delay on the Source: Tps = T_5-T_4

Processing Delay on the Destination: Tpd = T_3-T_2

Round Trip Time Delay: T = […] = T_2 - T_1 + T_4 - T_3

- We have no control of queuing delay on the source and destination, but this is experienced by real traffic too, and must be accounted as such

IPSLA Accuracy: UDP Echo Probe

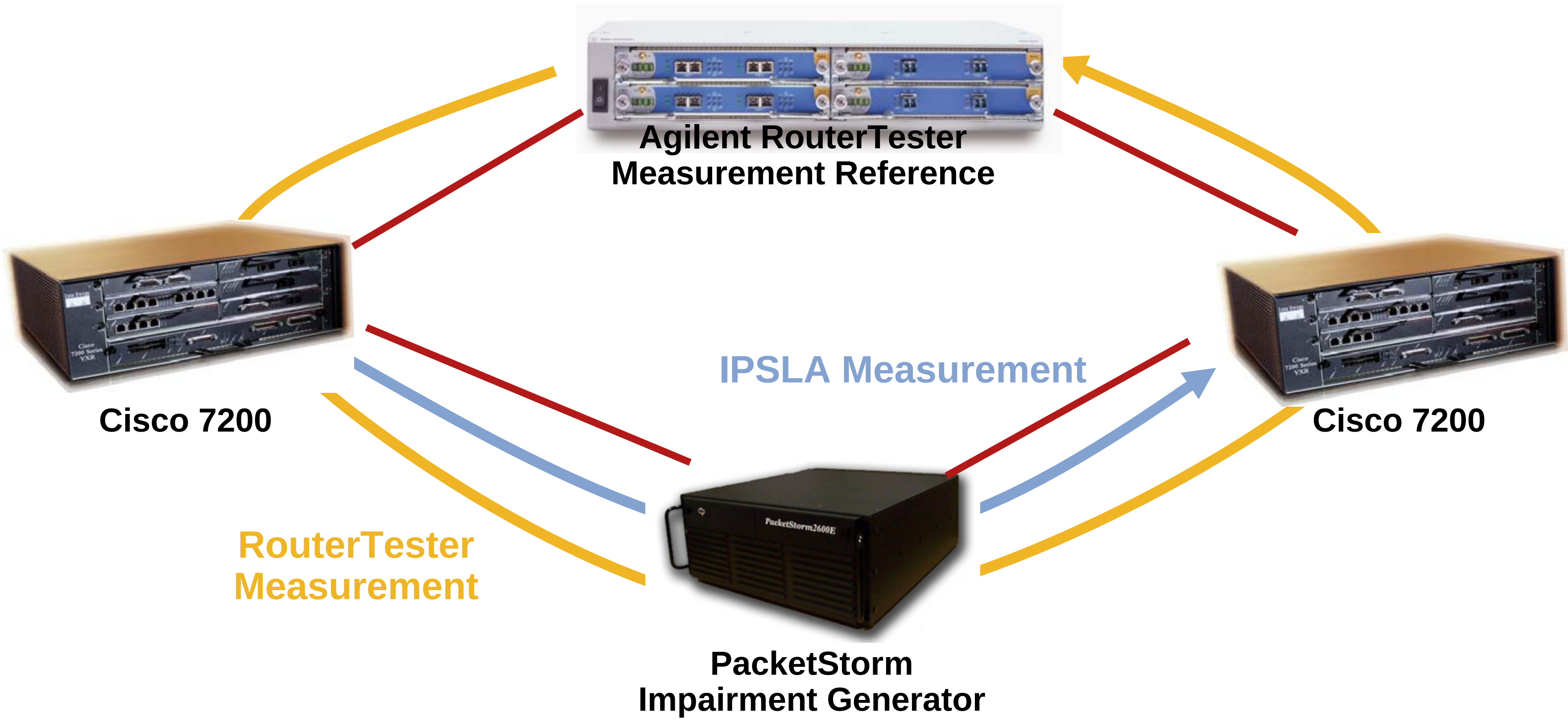
(90% Process Load)

- With unloaded receiver: 15.0 ms
- With 90% CPU receiver: 15.3 ms

The IPSLA Responder Processing Delay Will Be Subtracted from the Final Results

Absolute Accuracy Tests

- To validate IPSLA's accuracy, we wanted to compare its results with another measurement device
- We've used the following topology:



Test Results

- Release used: 12.3(7)T Advanced Enterprise on a Cisco 7200 VXR with NPE400
- RouterTester and IPSLA sending packets at the same rate
- All results obtained for delay and jitter are in sync with Agilent's result at +- 1 ms
- This was expected: it's the Cisco IOS timer granularity
- Spikes may happen during high-frequency interrupt events, like writing to NVRAM (write memory)

IPSLA Accuracy: The Dilemma

- A router is, basically, a forwarding machine
- IPSLA is a time sensitive application running on a forwarding machine
- Cisco IOS uses a non pre-emptive process scheduler
- This creates potential issues…
  but we have solutions

IPSLA Accuracy: ICMP vs. UDP

- As seen before—for RTT accuracy, always use UDP Echo or jitter with IPSLA responder
- Only in this case, processing time spent on the sender and responder routers will be subtracted
- Results more accurate regardless of the sender and receiver CPU process load
- But…if we have a high CPU interrupt load, like packet forwarding on centralized platforms, things may change…

IPSLA Accuracy: Test Results

- Accuracy is expected to be in the order of +/- 1 ms.
- Better accuracy is sometimes possible, but is dependant upon implementation details (hardware + IOS image + configuration).
- Process load has a negligible effect on UDP probes; remaining under 60% process load is a comfortable value

Accuracy with TOS-Marked Packets

- IPSLA probes can be sent with a specific Type of Service (TOS) value
- The right precedence will be applied when routing the packet, but what about the sending router?
- It depends…

Accuracy: Per Platform TOS Queuing

- Non-distributed platforms and 7500:

  Locally sourced traffic is processed by the service-policy on the outbound interface and will be queued according to the configured class-map/policy-map; so IPSLA packets will go through the corresponding configured queues

- Cisco 7600: Locally sourced traffic is queued onto a 'control queue' which has maximum weight therefore will not be dropped
- Cisco 10000 (ESR): Use 12.0(25)S or later
- Cisco 10720: from 12.0(25)S if an outbound service-policy is configured, locally sourced traffic is classified, but not marked
- For the Cisco 12000 (GSR):

  Make sure to use at least 12.0(28)S. For optimal results, leverage Unified QoS feature in 12.0(32)S where queuing is done in a consistent way across all LCs and RPs

Summary

- IP SLAs uses a special timestamping mechanism at interrupt level and its accuracy preserved even under high CPU load
- The absolute tested accuracy is +- 1 ms. In other words, when it says 35 ms, it could be somewhere between 34 ms and 36 ms.
- If the device is under high forwarding rate, or if it's a distributed platform, IP SLAs may lose some of its accuracy.
- If you can't live with that, we recommend the shadow router approach.

Agenda

- Reminder
- IPSLA Accuracy
- Performance and Scalability
- New Features
- Design Recommendations
- Get the Most Out of IPSLA
- IPSLA Initiative

Cisco IOS IP SLAs Performance: CPU Load by Platform

(Jitter Probe Running Eng 2—2000 Active Jitter Oper—Cisco IOS 12.3(3))

Oper/ Second	Oper/ Minute	2600	2620XM	3640	3725	7200VXR NPE225
4	240	14	7	6	2	4
8	480	20	8	9	3	3
12	720	29	12	13	2	3
16	960	35	15	17	3	3
20	1200	41	19	22	2	3
24	1440	48	24	25	3	3
28	1680	56	27	28	3	3
32	1920	63	28	31	2	4
36	2160	67	31	35	2	3
40	2400		34	38	3	7
44	2640		38	43	4	8
48	2880		42	47	5	8
52	3120		46	49	5	10
56	3360		48	43	6	11
60	3600		52	58	6	11

Cisco IOS IP SLAs Performance: CPU Load by Platform

(Jitter Probe Running Eng 2+—2000 Active Jitter Oper—Cisco IOS 12.4(PI3)T)

Oper/ Second	Pkts/ second	Oper/ Minute	2800	2811	2851	2691	3745	3845	3825	1841
4	200	240	3	3	1	2	1	0	2	3
8	400	480	6	5	2	3	1	1	3	4
12	600	720	8	7	3	4	2	2	5	6
16	800	960	10	9	4	5	2	2	7	8
20	1000	1200	13	11	4	6	3	3	8	10
24	1200	1440	15	13	5	8	4	4	10	11
28	1400	1680	18	14	6	9	4	4	12	13
32	1600	1920	20	16	7	10	5	5	14	15
36	1800	2160	23	18	8	11	5	6	16	17
40	2000	2400	24	20	9	12	6	6	17	18
44	2200	2640	27	21	10	14	7	7	19	20
48	2400	2880	29	21	11	15	7	8	21	22
52	2600	3120	32	22	12	16	8	8	23	23
56	2800	3360	34	22	13	17	9	9	26	24
60	3000	3600	36	23	14	18	9	9	27	26

IP SLAs Typical Memory Usage

	Eng2 12.2(13)T
UDP Jitter	< 14KB
UDP Echo	< 3.5KB
ICMP Echo	< 3.2 KB

Summary

- Under normal conditions and with reasonable targets, a performance issue with IP SLAs is unlikely
- Memory usage is reasonable, and should never be a problem on any platform.
- Compared to Engine 1, both performance and memory usage have been improved on IPSLA Engine 2 and 2+

Agenda

- Reminder
- IPSLA Accuracy
- Performance and Scalability
- New Features
- Design Recommendations
- Get the Most Out of IPSLA
- IPSLA Initiative

IP SLAs RTP VoIP -- The Problem

- How to evaluate the clarity of a voice call?
- Existing operations measures at the IP level, but have no idea on how call clarity is impacted.
- We move toward *service-oriented SLAs*, and therefore looking at the application impact rather than at the pure IP metrics.

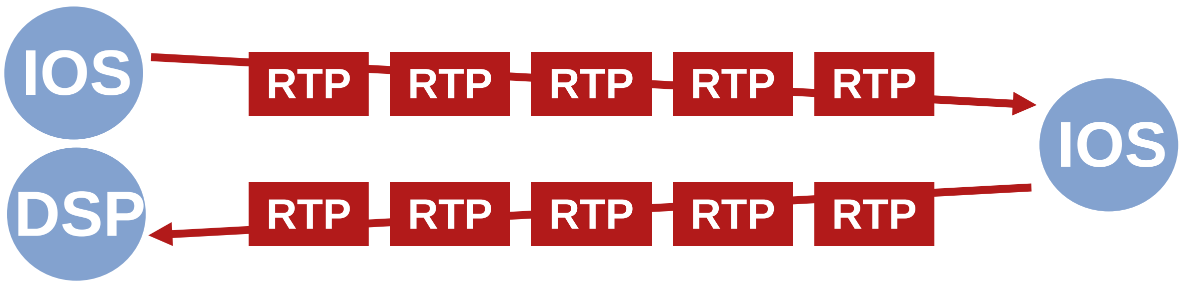
Understand the "service" notion...

- You are telling this (click to hear):
- With 25 ms jitter, 1% drop and 1% packet reordering, you will hear:
- With 30 ms jitter, 5% drop and 2% packet reordering, you will hear:
- This is why the notion of "service" is important: it is hard to predict how a particular network impairment will influence the underlying protocol and service.

*** Actual sound files and samples are courtesy of InterWorking Labs, Inc.**

The RTP Operation

- Sends a real RTP stream, generated in software.
- Is is received and decoded by a real Digital Signal Processor (DSP).
- The jitter and drop metrics will be measured directly in the DSP, in hardware.
- We do support two DSPs, on a variety of platforms.



Collected Set of Statistics

- As of today, the IP SLAs RTP VoIP Operation can measure and report the following metrics:

    RFC1889 (RTP) inter-arrival Jitter at source and destination

    R-factor at source and destination

    MOS-CQ (Mean Opinion Score -- Conversation Quality) estimated value using R factor and G.107 R-factor to MOS conversion table.

    Packet Loss at source and destination

    Network round trip time

    Early packets

    Packets Out of Sequence

    Late Packets

R-Factor from ITU-G107 (E-Model)

$$R = Ro - Is - Id - Ie,eff + A$$

- Ro — Noise
- Is — Loudness
- Id — Echo and delay
- Ie,eff — Distortion/ discontinuity

Advantage factor
Complements the effect of the convenience of mobile or other communication on a subjective quality (satisfaction).

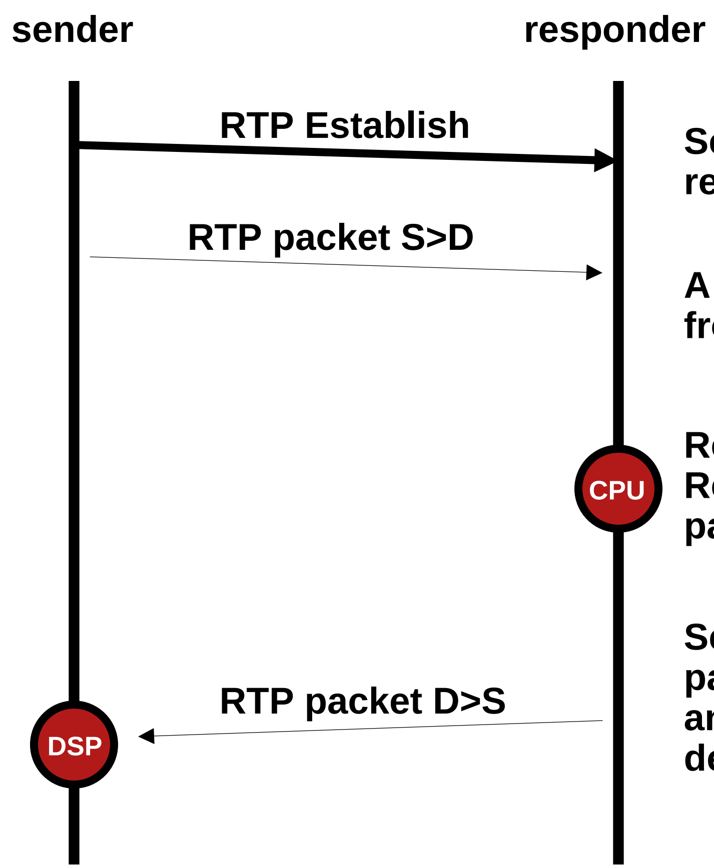
Equipment impairment factor
Represents subjective quality impairments due to low bit rate CODEC, packet/cell loss, etc.

Delay impairment factor
Represents subjective quality impairments due to talker echo, listener echo, and absolute delay

Simultaneous impairment factor
Represents subjective quality impairments due to OLR (loudness), sidetone, and quantization distortion.

Basic signal-to-noise ratio
Represents subjective quality impairment due to circuit noise, room noise at sending and receiving sides, and subscriber line noise.

IP SLAs RTP VoIP Operation Flow



Sender establishes an RTP session with the responder, as well as the IP SLAs control phase.

A real RTP stream is generated in software and sent from the sender to the responder.

Responder is decoding the stream in software. Receive and transmit timestamps are added in the RTP packet payload, and sent back.

Sender uses its DSP to receive the stream. The payload contains timestamps encoded at destination and is used for source to destination IP metrics, while destination to source are DSP frame metrics.

Cisco IOS Version Support

- Platforms supported: 175x, 2600, 2800, 3600, 3800, 7200 running 12.4(4)T “IP Voice” or higher.
- In the original release, one does only measure in one direction: responder to sender.
- The bi-directional operation was introduced in 12.4(6)T.

IP SLAs RTP VoIP -- Config Example

Example of configuration (sender):

```
controller E1 0/0
 ds0-group 15 timeslots 3 type e&m-wink-start

ip sla 3
 voip rtp 10.48.164.20 source-voice-port 0/0:15 codec g711ulaw
ip sla schedule 3 start-time now
```


IP SLAs RTP VoIP -- Output

- Output example:

```
etychon-s#sh ip sla sta 3 details

Round Trip Time (RTT) for       Index 3
Type of operation: rtp
Latest operation start time: 07:24:11.941 UTC Mon Feb 27 2006
Latest operation return code: OK
Latest RTT (milliseconds): 0
Source Measurements:
        Interarrival Jitter: 0
        Packets Lost: 0         Packets OutOfSequence: 0
        Packets Late: 0         Packets Early: 0
        R-factor: 92    MOS-CQ: 4.34
Over thresholds occurred: FALSE
Operation time to live: Forever
Operational state of entry: Active
Last time this entry was reset: Never
```


IP SLAs RTP VoIP -- Bidir Output

- Example:

```
etychon-s#sh ip sla sta 100 det

Round Trip Time (RTT) for       Index 100
Type of operation: rtp
Latest operation start time: 08:08:26.345 UTC Thu Feb 9 2006
Latest operation return code: OK
Latest RTT (milliseconds): 1
Source to Destination Path Measurements:
        Interarrival Jitter: 0
        Packets Sent: 500
        Packets Lost: 0         Packets MIA: 0
        Estimated R-factor: 92  MOS-CQ: 4.34
        One way latency(avg/min/max): 1/0/1
Destination to Source Path Measurements:
        Interarrival Jitter: 0
        Packets Sent: 501
        Packets Lost: 0         Packets OOS: 0
        Packets Late: 0         Packets Early: 0
        Estimated R-factor: 92  MOS-CQ: 4.34
        One way latency(avg/min/max): 0/0/1
Over thresholds occurred: FALSE
Operation time to live: Forever
Operational state of entry: Active
Last time this entry was reset: Never
```

IP SLAs RTP VoIP -- Status and DSP Type

- Check active RTP streams:

```
etychon-s#show voip rtp connections detail
VoIP RTP active connections :
No. CallId  dstCallId  LocalRTP RmtRTP LocalIP       RemoteIP
1   123     0          19296    19232  10.48.164.19  10.48.164.20
 callId 123 (dir=2): called= calling= redirect=
   1 context 81F5B8E0 xmitFunc 812BA3FC
   2 context 81F5B8E0 xmitFunc 812BA3FC
Found 1 active RTP connections
```

Check DSP Type installed:

```
etychon-s#sh voice dsp

DSP  DSP             DSPWARE CURR  BOOT                               PAK     TX/RX
TYPE NUM CH CODEC    VERSION STATE STATE   RST AI VOICEPORT TS ABORT  PACK COUNT
==== === == ======== ======= ===== ======= === == ========= == ===== ============
C549 000 00 g711ulaw  4.1.43 Idle  Idle      0  0 3/0       NA     0          9/3
C549 000 01 g711ulaw  4.1.43 Idle  Idle      0  0 3/1       NA     0          9/3
```

Multi-Operation Scheduler

- Avoid overloading the router at boot time with all operations starting at the same time. We introduce the notion of group.
- Introduced in 12.3(8)T, it lets you start many operations at once, with automatic smooth "start-time".
- Example, start operations 1 to 10 within the next 10 seconds:

```
r1(config)#ip sla group schedule 1 1-10 schedule-period 10 \
             start-time now
r1#sh ip sla op | i start
Latest operation start time: *12:50:51.599 PST Mon Apr 18 2005
Latest operation start time: *12:50:52.599 PST Mon Apr 18 2005
Latest operation start time: *12:50:53.599 PST Mon Apr 18 2005
Latest operation start time: *12:50:34.579 PST Mon Apr 18 2005
Latest operation start time: *12:50:35.579 PST Mon Apr 18 2005
Latest operation start time: *12:50:36.579 PST Mon Apr 18 2005
Latest operation start time: *12:50:37.579 PST Mon Apr 18 2005
Latest operation start time: *12:50:38.579 PST Mon Apr 18 2005
Latest operation start time: *12:50:39.579 PST Mon Apr 18 2005
Latest operation start time: *12:50:40.591 PST Mon Apr 18 2005
```


Randomized start-time

- The group start time can be randomized. This avoids a “synchronization effect” (ie: the test happens always at the same time something else is happening, like a route update).
- This example starts operation 1 to 5 within the next 44 seconds, and each operation will have a random frequency varying between 10 and 15 seconds.

```
ip sla group schedule 1 1-5 schedule-period 44 frequency range 10-15 start-
   time now life forever

etychon-1#sh ip sla op | i start
Latest operation start time: *12:56:12.243 PST Thu Oct 13 2005
Latest operation start time: *12:56:06.323 PST Thu Oct 13 2005
Latest operation start time: *12:56:07.743 PST Thu Oct 13 2005
Latest operation start time: *12:56:13.323 PST Thu Oct 13 2005
Latest operation start time: *12:56:08.643 PST Thu Oct 13 2005
etychon-1#sh ip sla op | i start
Latest operation start time: *13:00:19.423 PST Thu Oct 13 2005
Latest operation start time: *13:00:15.895 PST Thu Oct 13 2005
Latest operation start time: *13:00:21.015 PST Thu Oct 13 2005
Latest operation start time: *13:00:25.303 PST Thu Oct 13 2005
Latest operation start time: *13:00:14.635 PST Thu Oct 13 2005
```


NTP Awareness

- Starting 12.3(14)T the IP SLAs process is aware of NTP state: an acceptable offset tolerance can be configured per operation
- This prevents inaccurate measurement when NTP is desynchronized
- One-way results are ignored if the NTP offset is over the threshold (in microseconds)

```
etychon-s#sh ntp assoc

      address         ref clock     st  when  poll reach  delay  offset    disp
*~10.0.0.2         127.127.7.1       8     0    64    17     4.0  -2.00  1875.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

```
ip sla 1
 udp-jitter [...]
 clock-tolerance ntp oneway absolute 2500
```

New CLI

- Will be introduced in four phases, removing the obsolete "`rtr`" name for "`ip sla`".

  Phase I in 12.3(12)T: Basically a replacement of "rtr" by "ip sla monitor".

  Phase II in 12.3(14)T: few changes in the show commands, config unchanged.

  Phase III in 12.4(6)T: removal of "monitor" and simplified in-operation syntax.

  Phase IV: template-based configuration thanks to the integration with the modular QoS command line. (Work in progress)

IP SLAs -- MPLS Health Monitor

- Automatically create and delete IP SLAs LSP ping or LSP traceroute operations based on network topology
- Works on the MPLS L3 layer, under the IP layer. Discovers MPLS issues even when IP routing is working ok.
- Dramatically reduces troubleshooting time, and cost associated to maintenance of MPLS networks.
- Other PEs are discovered using BGP next-hop, and operations configured accordingly.
- Requires 12.2(27)SBC and later.

Agenda

- Reminder
- IPSLA Accuracy
- Performance and Scalability
- New Features
- Design Recommendations
- Get the Most Out of IPSLA
- IPSLA Initiative

“Reasonableless Test”

- Don’t overdo it, your metrics must be:
    - Attainable
    - Measurable
    - Relevant
    - Controllable
    - Mutually Acceptable
    - Understandable
    - Cost Effective
- Use a limited but relevant number of indicators.
- Better is the enemy of good: good is good enough.

Typical SLA Requirements

Traffic Type	Maximum Packet Loss	Maximum One-Way Latency	Max. Jitter
VoIP (land line quality)	1 %	120 ms	30 ms
Video-conferencing	1 %	200 ms	50 ms
Streaming video (one way video)	2 %	5 s	N/A (assuming the receive buffer is large enough)

Real-time vs. Periodic Reporting

Real-Time Reporting	Periodic Reporting
▪ Confirmation of status ▪ Potential problems ▪ Notification ▪ Nature of problem	▪ Historical reports ▪ Objectives vs. Estimates ▪ Anticipation: potential impact, things to avoid ▪ Change in service levels

Cisco IOS IPSLA Uses and Metrics

	DATA TRAFFIC	VoIP	SERVICE LEVEL AGREEMENT	AVAILABILITY	STREAMING VIDEO
REQUIREMENT	• Minimize packet loss • Maximize bandwidth • Verify Quality of Service (QoS)	• Minimize delay, packet loss, jitter	• Measure delay, packet loss, jitter • One-way	• Connectivity testing	• Minimize delay, packet loss
IPSLAs MEASURMENT	• Packet loss • Latency • per QoS	• Jitter • Packet loss • Latency • MOS Voice Quality Score	• Jitter • Packet loss • Latency • One-way • Enhanced accuracy • NTP	• Connectivity tests to IP devices	• Jitter • Packet loss • Latency

Class of Service

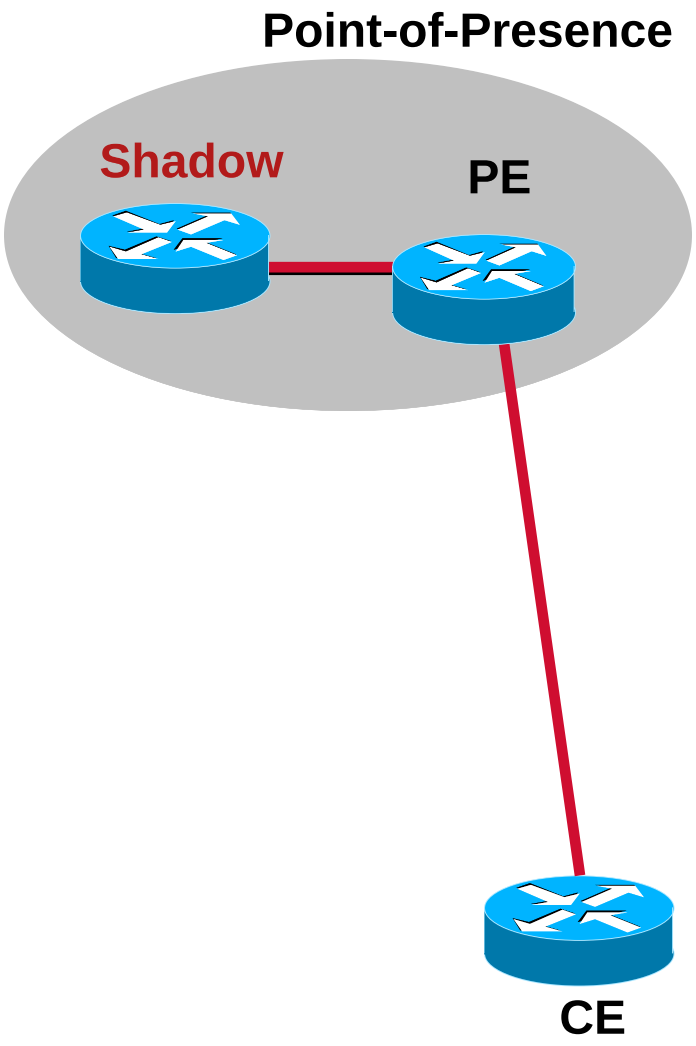
- One operation instance to measure each class of service
- Same operation type for all classes
- Traffic coloring from within IP SLA with TOS/DSCP configuration
- Bear in mind the corner case with locally generated and colored traffic on some distributed platforms
- Workaround is to use a Shadow Router

Why Using a Shadow Router?

- A *shadow router* is a dedicated box for IP SLAs... But why?
- If your Provider Edge (PE) router is already overloaded (> 60% CPU at interrupt level)
- If your PE lacks memory
- If your PE is a distributed platform
- If you want to separate measurement from forwarding
- Upgrade freely for the the latest and greatest IP SLA features without disturbing the traffic, then…
- Use a shadow router (router dedicated to IPSLA)

Shadow Router Configuration

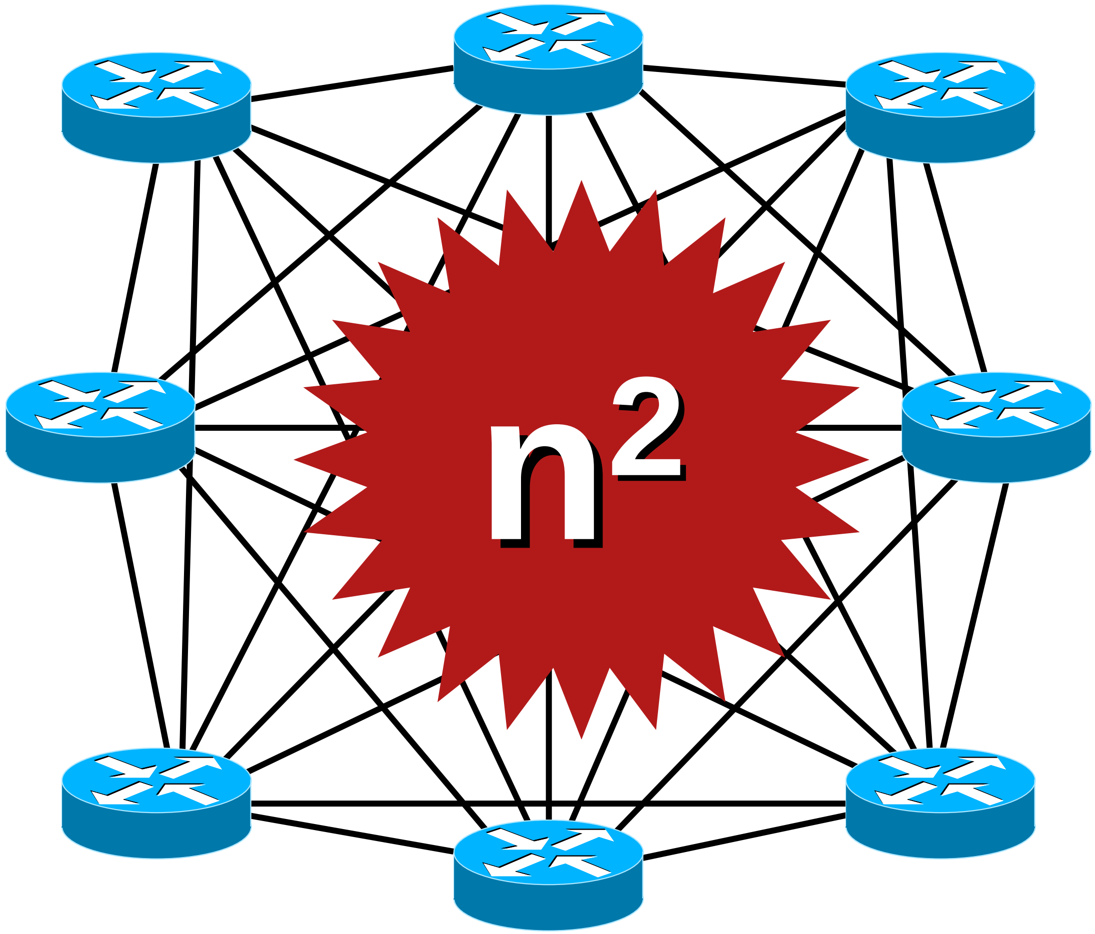
- A shadow router is typically a dedicated router located near a ideal measurement point.
- A point-of-presence (POP) is an ideal location.
- It can be connected to the PE via various methods: direct IP connection, tunnels, dot1q,…



How to Probe?

- Full mesh
- Full mesh between same-customer CPEs
- Partial mesh
- Composite SLAs

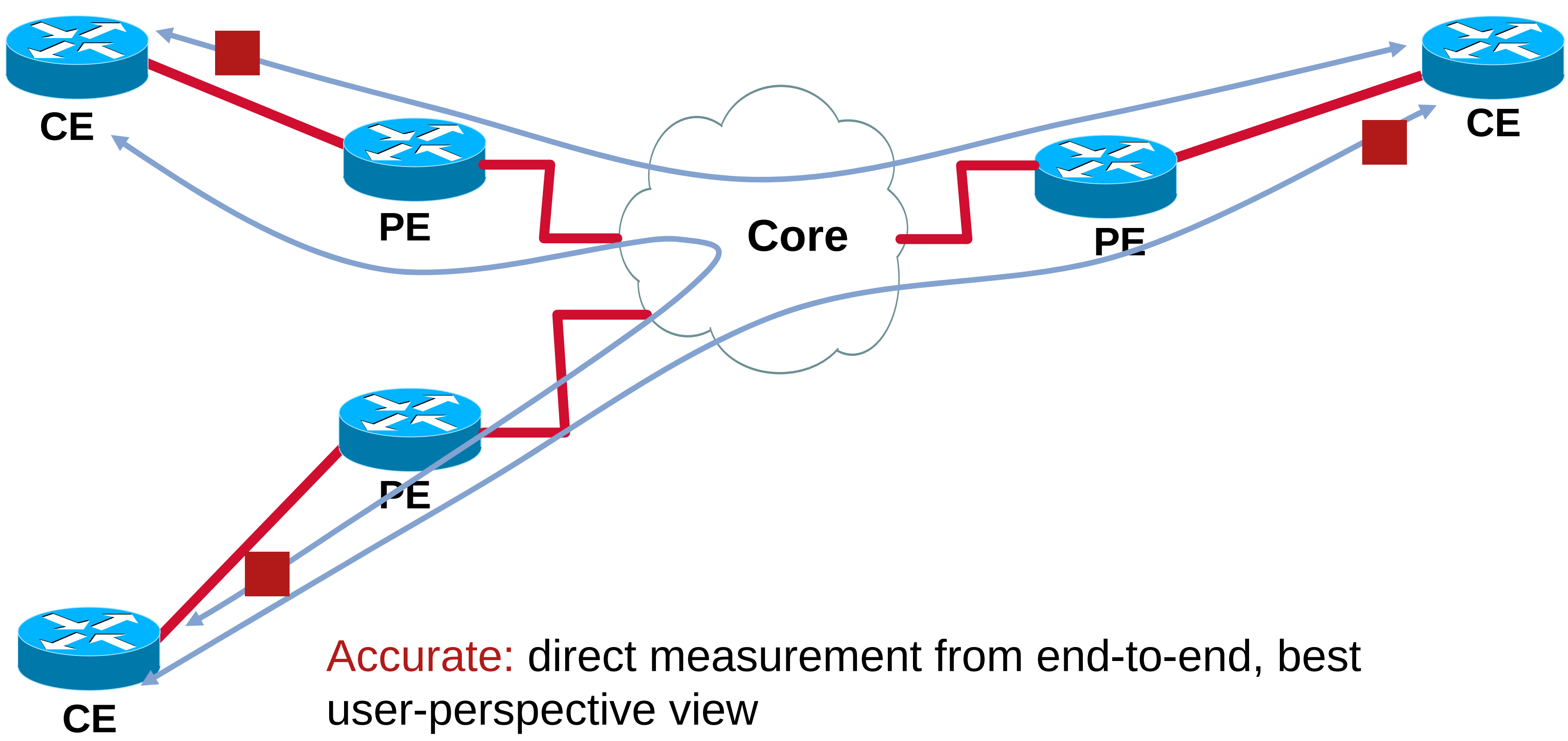
Full Mesh



Nodes	Operation
2	1
3	3
4	6
5	10
6	15
7	21
8	28
…	…
100	4950

- **Number of operations is proportional to the square of the number of nodes**
- **Does not scale**

Full Mesh CE-to-CE [Example]

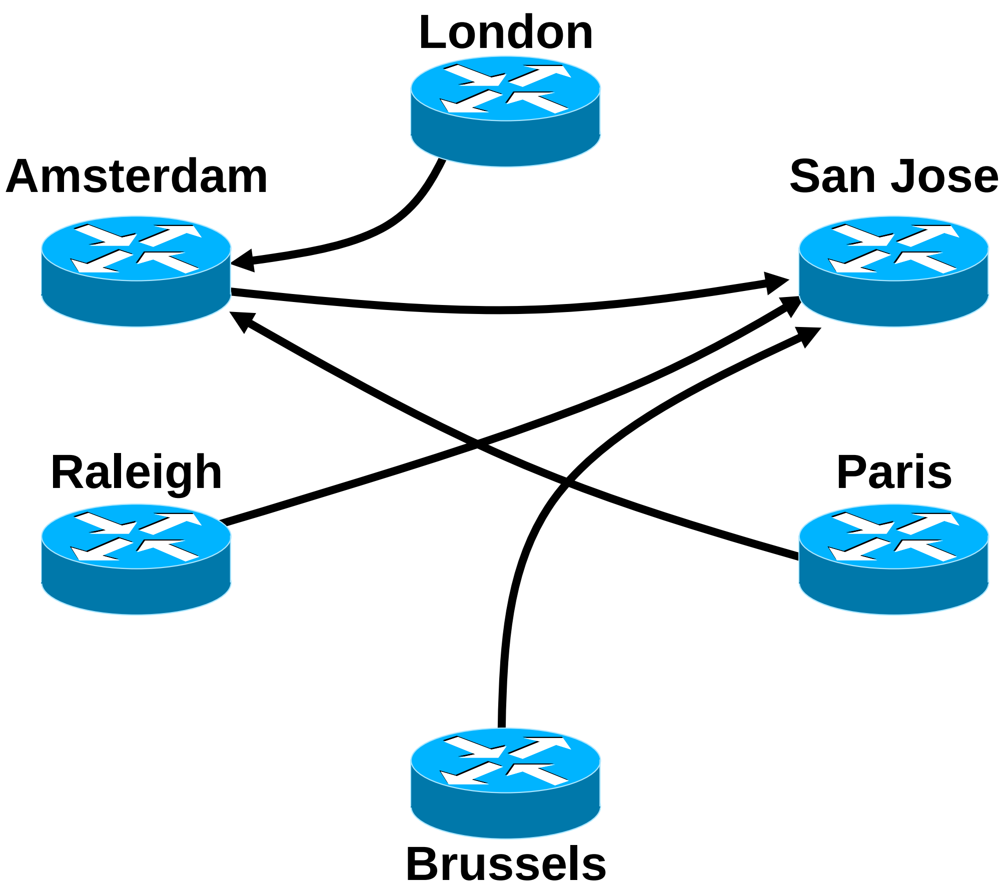


Accurate: direct measurement from end-to-end, best user-perspective view

Expensive: for n nodes, requires n(n-1)/2 operations

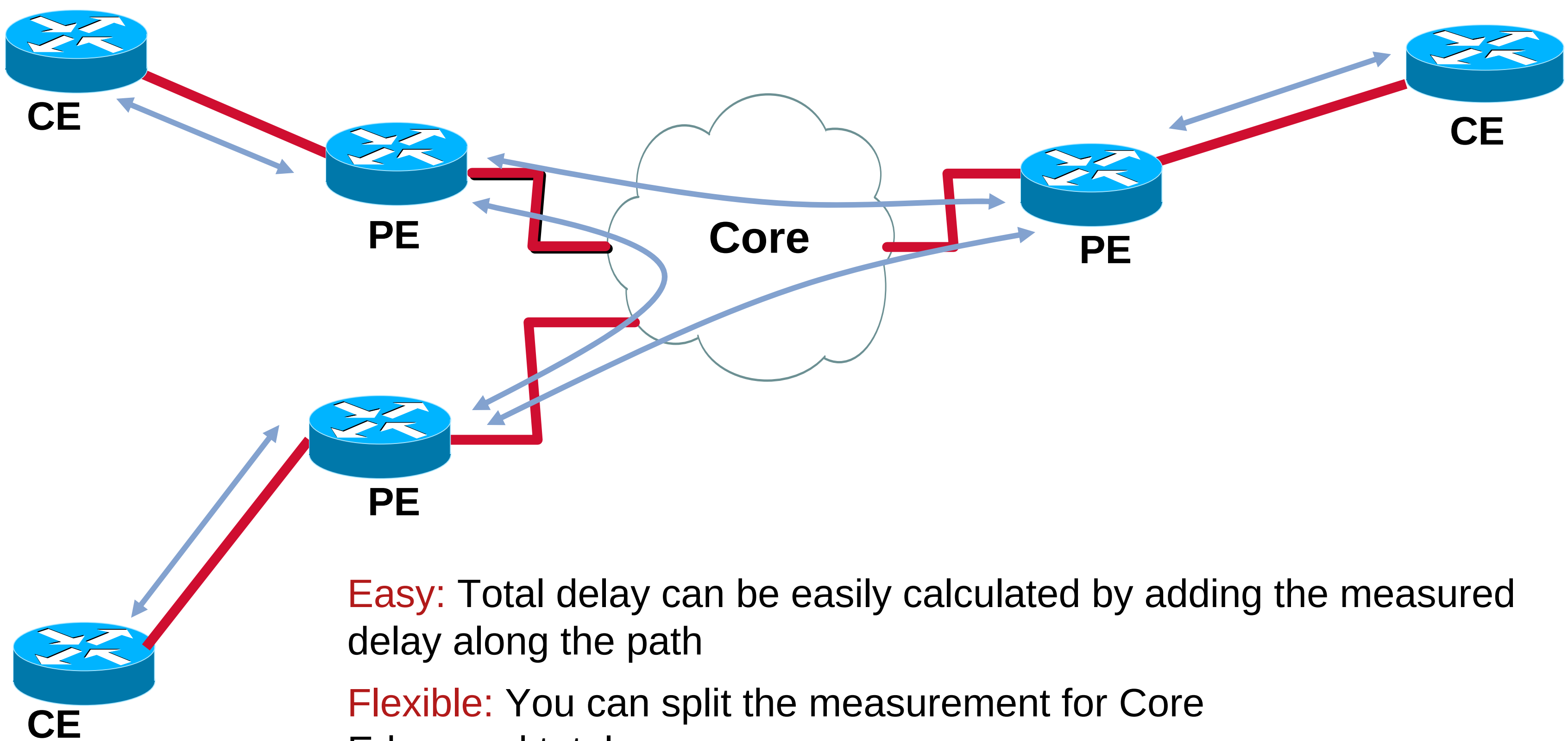
In certain cases, it might be difficult to poll the results with SNMP on the CE

Partial Mesh



- Full mesh is not always desirable, while partial mesh dramatically reduces the number of operations.
- Measurement points can be based on **traffic matrix**, traffic importance,...
- For instance, try a coverage objective for 80% of the traffic
- To build a traffic matrix, use NetFlow.

Composite SLA for Delay [Example]



Easy: Total delay can be easily calculated by adding the measured delay along the path

Flexible: You can split the measurement for Core Edge, and total

Measurements are less accurate, as each measurement carry its own error tolerance (typically ± 1 ms per measurement)

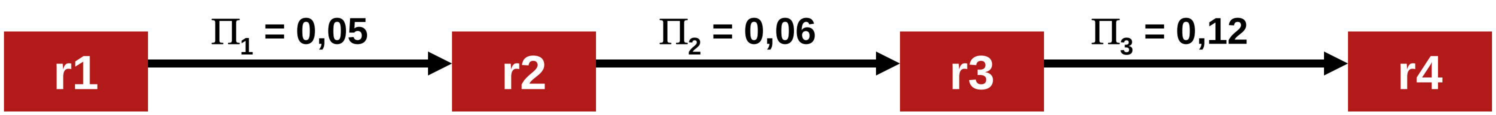
Composite SLA for Packet Drop [1/2]

- A trivial solution might is to consider the sum of drop probabilities; this is conservative
- A more accurate approach is to invert the probability of a successful packet delivery
- If Π_x is the loss probability across section x, then the total loss probability is:

$$\Pi_{1 \ldots x} = 1 - [(1 - \Pi_1).(1 - \Pi_2)\Lambda\ (1 - \Pi_n)]$$

Composite SLA for Packet Drop [2/2]

Example: We Have Three Sections with Various Drop Probabilities:



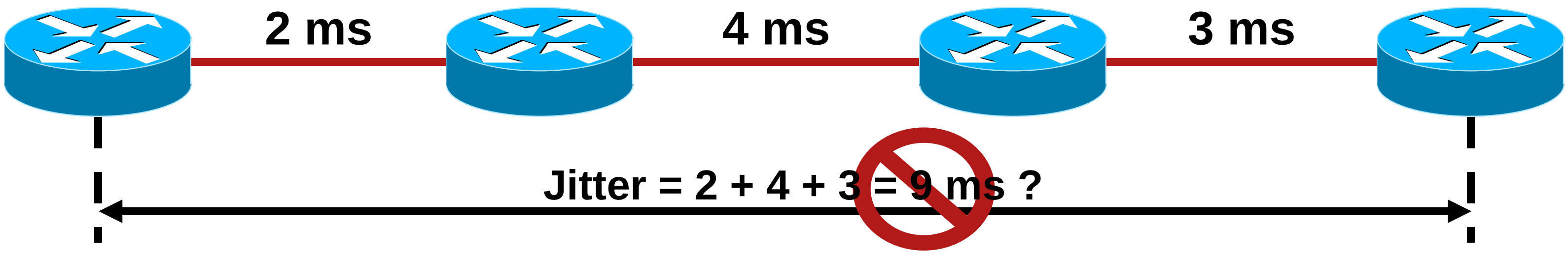
- **First solution (approximation):**

  0,05+0,06+0,12=0,23 (23%)

- **Second solution (exact):**

  1-[(1-0,05)x(1-0,06)x(1-0,12)]=0,21416 (21,4%)

Composite SLA for Jitter



Can We Add a Jitter Value to a Jitter Value?

- Short answer: NO!
- This is not a valid approach to calculate total jitter based on measured jitter, because we don't know how to do it… (jitter is not additive)
- Too many factors: positive jitter, negative jitter, percentile-95 of jitter, average jitter,…
- You'd better measure it, not calculate it

Summary

- PE-PE, PE-CE or CE-CE, full-mesh or partial-mesh is all your decision!
- IPSLA can run on almost any existing Cisco router. When this is not possible/desirable then a shadow router is recommended
- Composite SLAs are a good idea while end-to-end jitter results are not required

Agenda

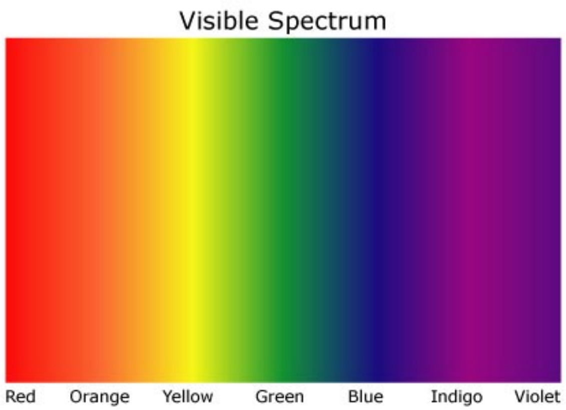
- Reminder
- IPSLA Accuracy
- Performance and Scalability
- New Features
- Design Recommendations
- Get the Most Out of IPSLA
- IPSLA Initiative

Common Questions…

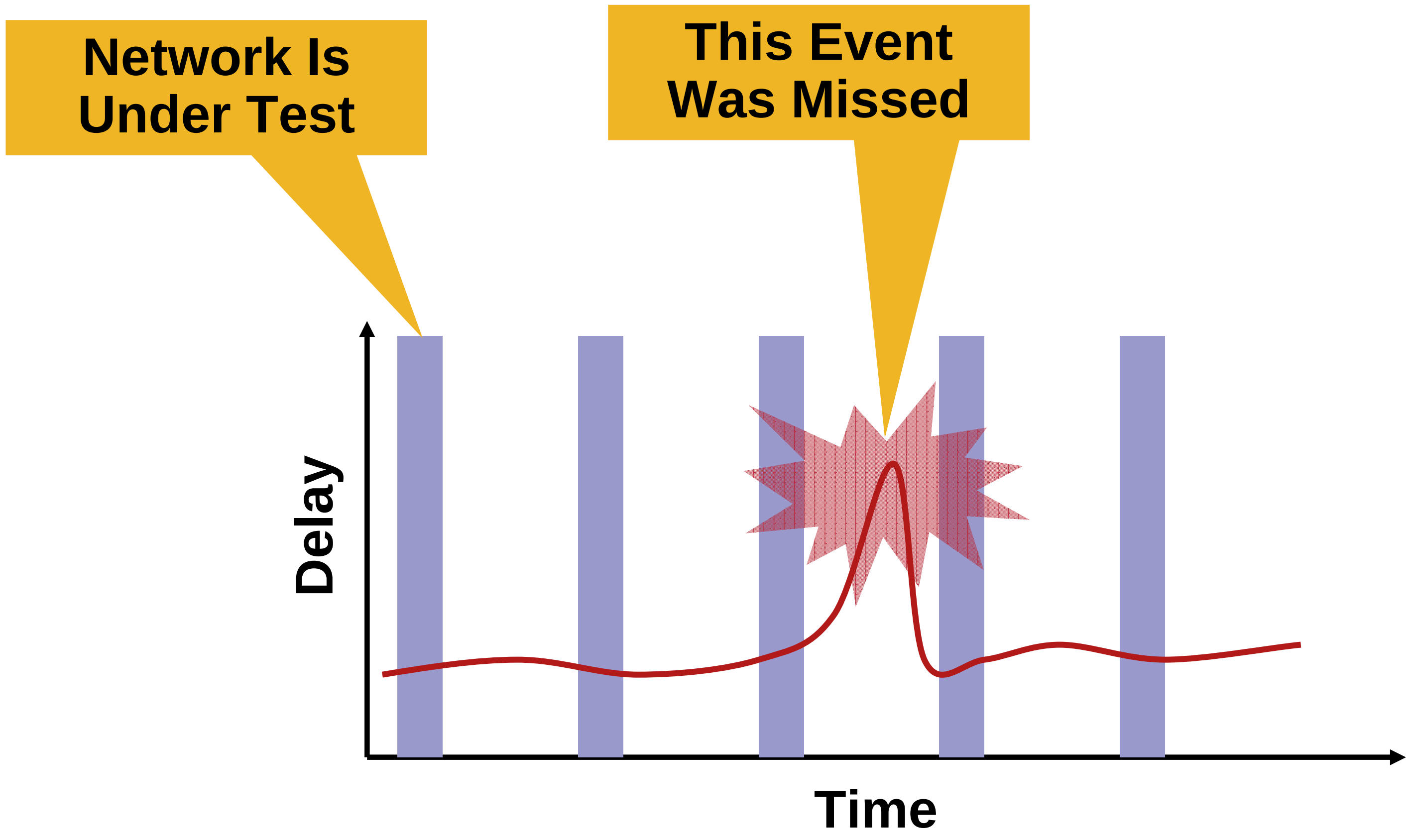
- How should I configure my operations to accurately measure jitter/delay/packet loss?
- How many packets should be sent per operation?
- How frequently?
- What percentage of by bandwidth should be dedicated for measurement?

Spectrum of Test

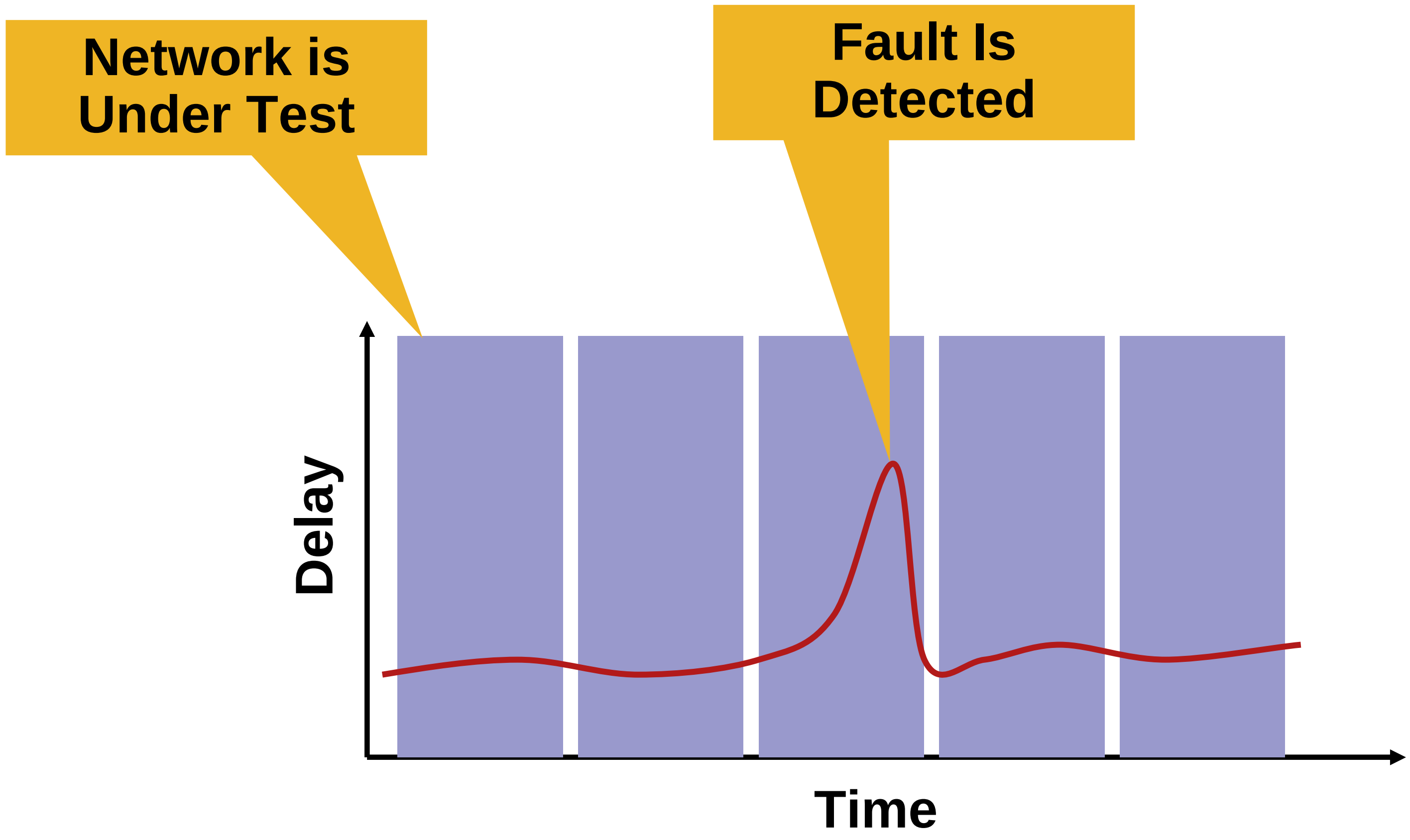
- This is the proportion of time during which the network is under test
- A small spectrum of test means a small probability of catching an event
- For example: running a test for 20 seconds every 60 seconds is equivalent to a 33% spectrum of test



Spectrum of Test

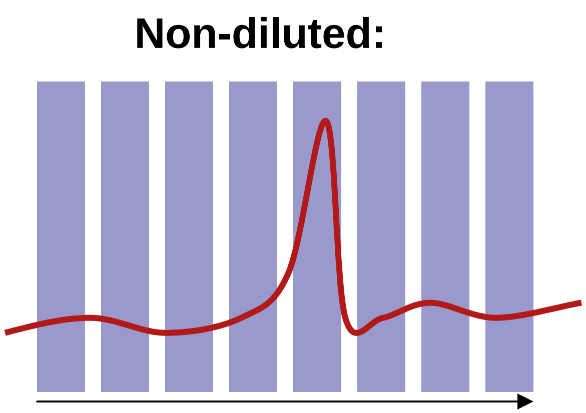


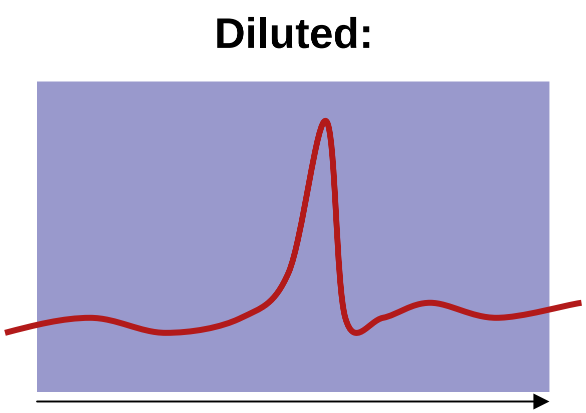
Spectrum of Test



Number of Packets

- The more packets sent:
    - The larger the population
    - The more diluted are the results
- At identical frequency, the longer the operation, and the wider the test spectrum.
- Example of result dilution with the same spectrum, but a bigger number of packets per operation.





Frequency

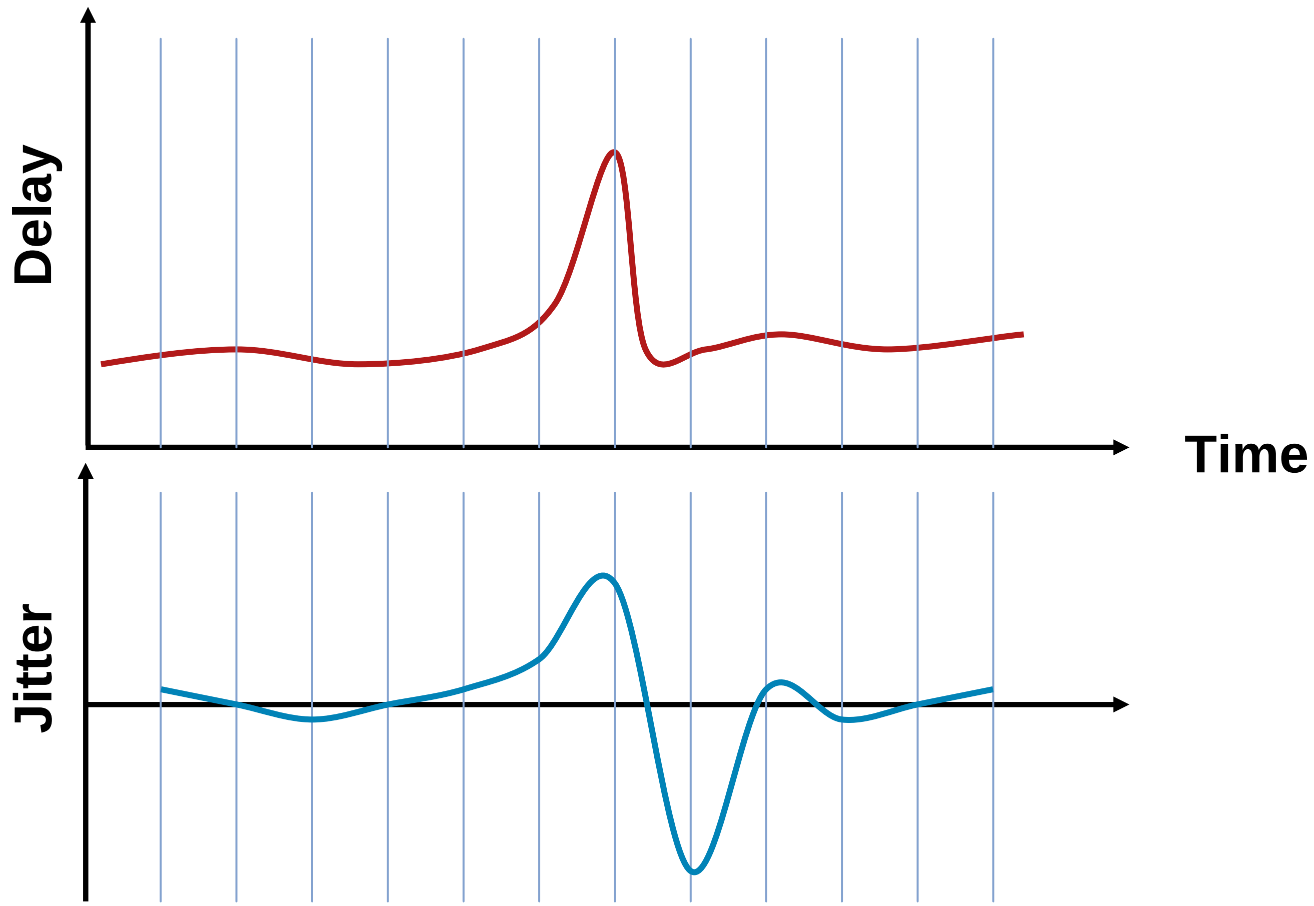
- The operation frequency, as well as operation duration, have a direct impact on the SPECTRUM OF COVERAGE
- Increasing the frequency will increase your spectrum of coverage, and increase the bandwidth consumed but will not change the accuracy

Interval

- The interval is the space between two consecutive probe packets
- Long intervals (hundreds of ms) are for trends, and will lead to higher jitter results
- Short intervals (low tens of ms) are for very precise measurement, limited in time; the jitter is expected to be smaller in that case

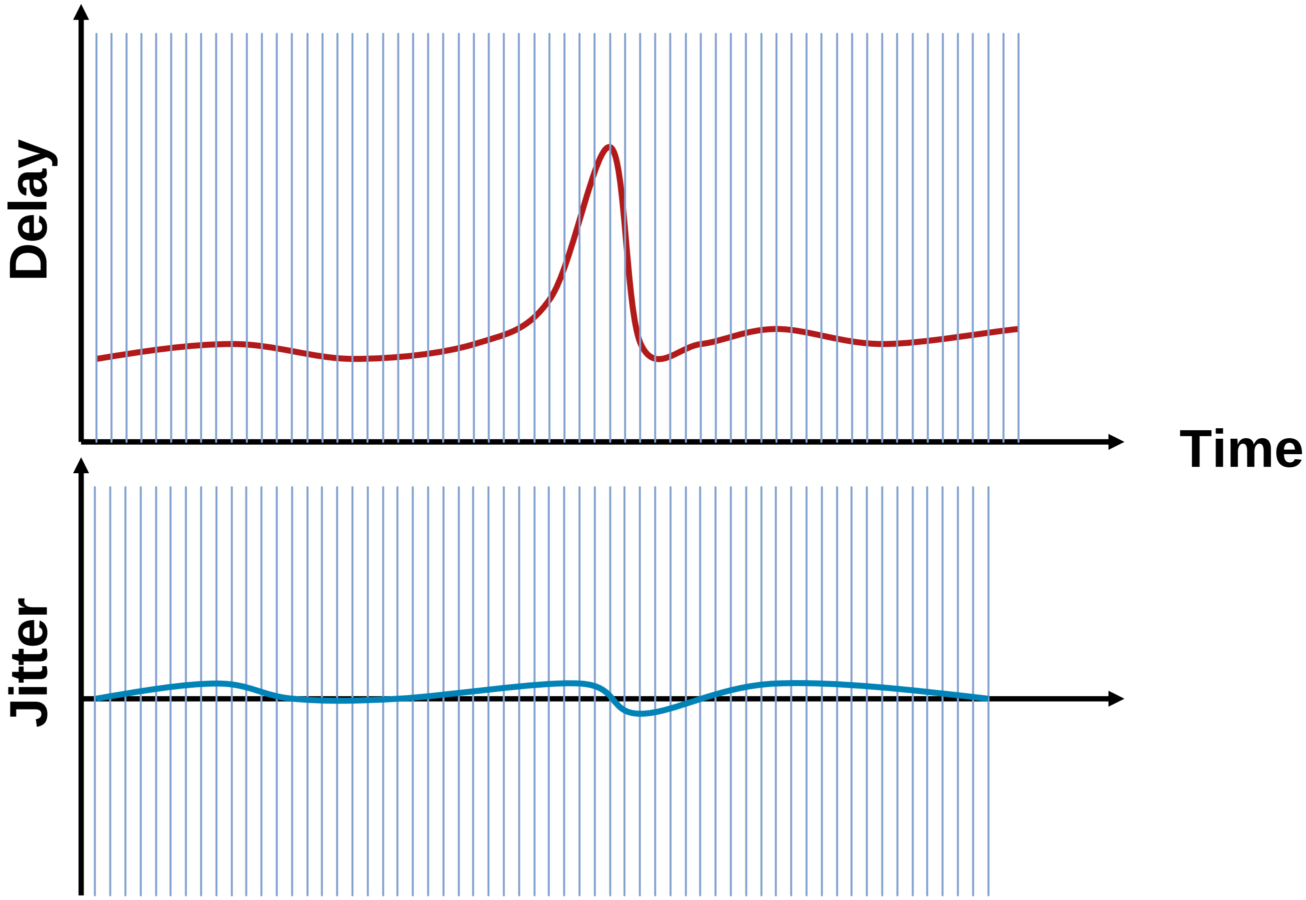
Interval Effect of Jitter

- Longer intervals ultimately measures bigger jitter, because of coarse granularity:



Interval Effect of Jitter

- Shorter intervals measurements are more granular, and hence give less jitter:



Interval and Jitter

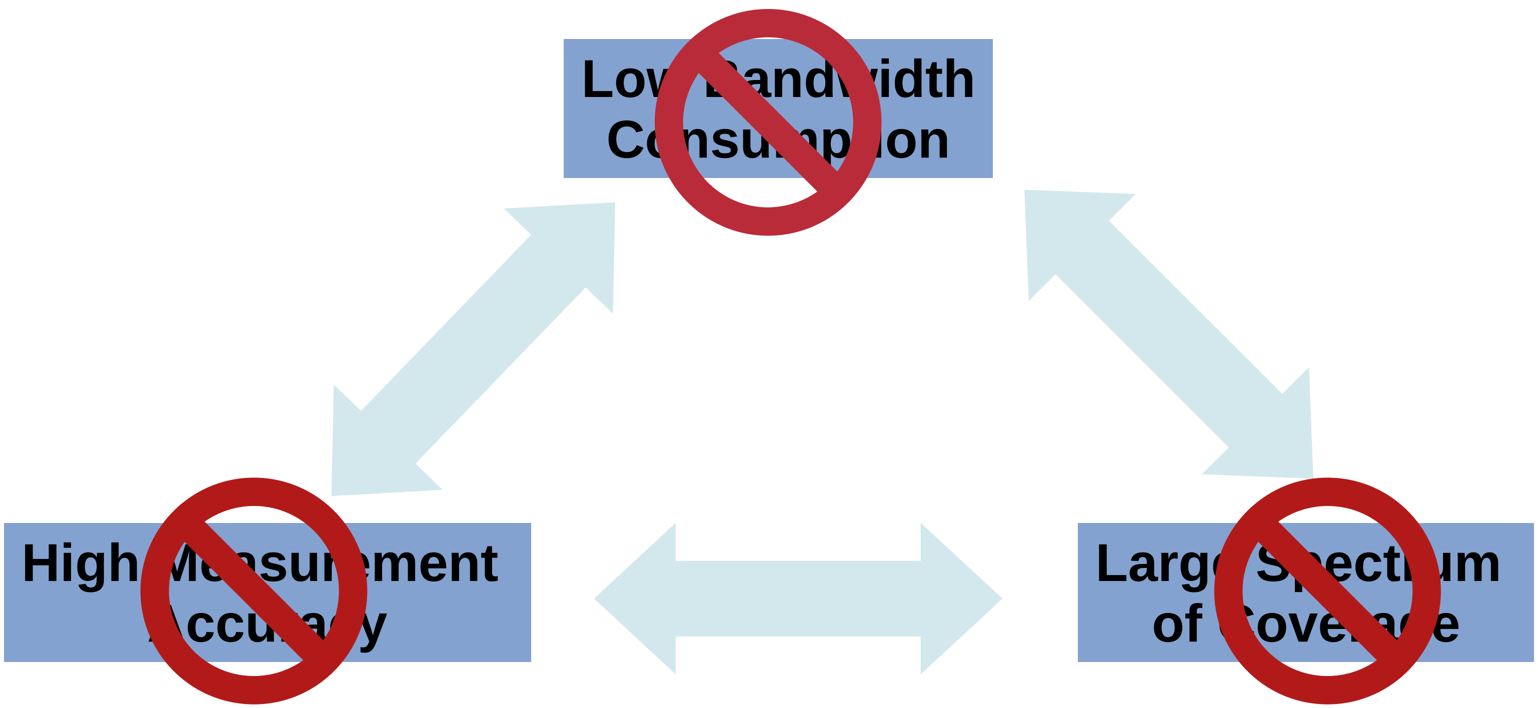
- Compare different jitter measurements ONLY if the measurement intervals are identical
- Short interval is more accurate, but more expensive: use it occasionally to have a true application-like jitter
- Long interval is less accurate, but consumes less bandwidth: use it to expand your test spectrum and keep an eye on your jitter trends

Packet Size

- The main effect of packet size is to modify the SERIALIZATION DELAY
- On fast links, this is negligible compared to the propagation delay, so the packet size has little or not effect but to consume bandwidth
- Use small packets of fast links, like on core network
- Use realistic packets for low-speed access links, where the serialization delay is a factor we need to count

Summary

- The design will have to accommodate some tradeoffs, you can choose two out of three:



Agenda

- Reminder
- IPSLA Accuracy
- Performance and Scalability
- New Features
- Design Recommendations
- Get the Most Out of IPSLA
- IPSLA Initiative

Cisco IOS IP SLAs Partners

Cisco Network Management Solution	
IP Communications Service Monitor	Telephony Monitoring
Internetworking Performance Monitor	Enterprise performance measurements

THIRD PARTY PRODUCTS

hp invent

concord | ca

InfoVista

IBM

MICROMUSE NETCOOL SOLUTIONS An IBM Company

New Partners 2006

NetQoS Performance Experts

AdventNet Excellence Matters

wiredcity a division of OSIsoft, Inc.

Agilent Technologies

CRANNOG SOFTWARE

References

- Cisco IOS IPSLA home page

  http://www.cisco.com/go/ipsla

- For questions related to Cisco IP SLAs that cannot be handled by the Technical Assistance Center (TAC), feel free to write an email to:

  ask-ipsla@cisco.com

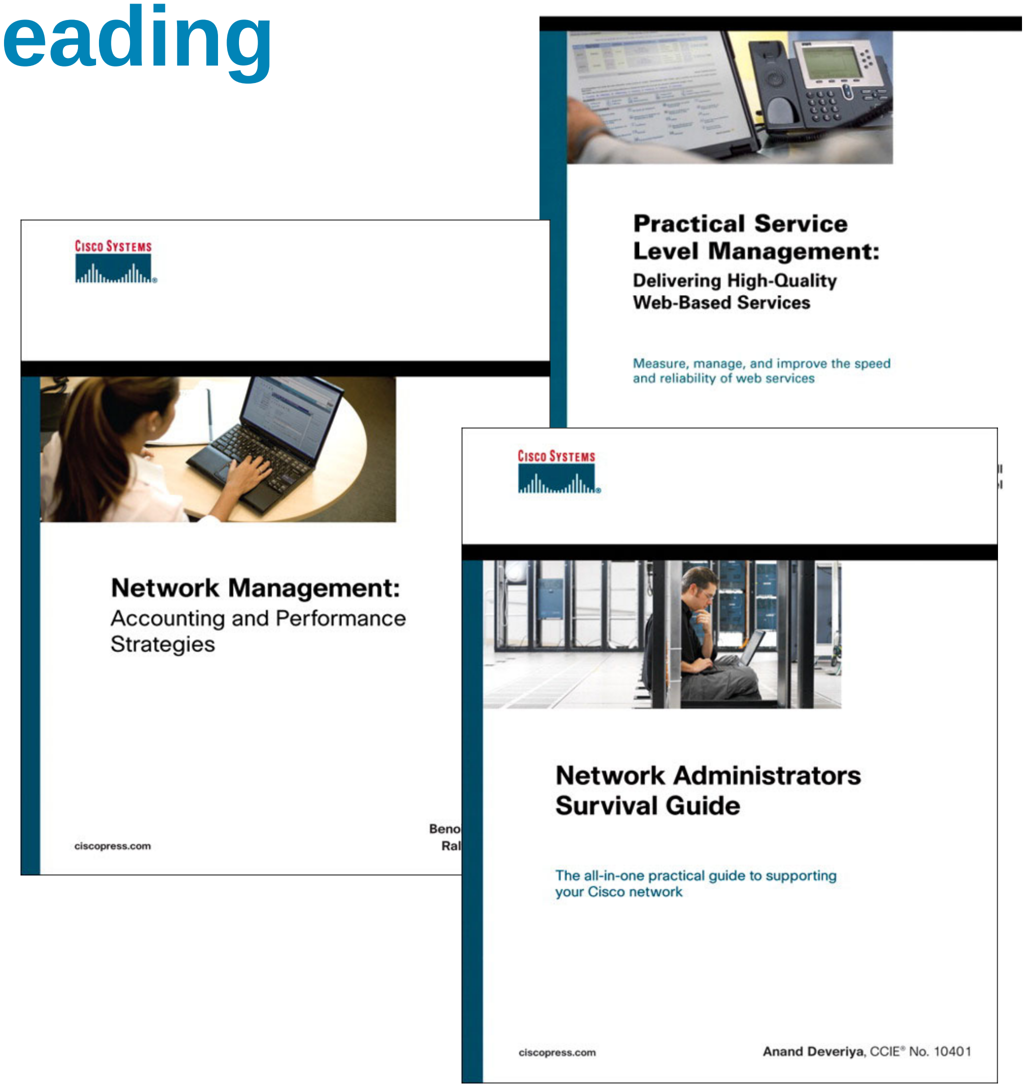
Summary & Conclusion

- IPSLA is a Cisco IOS feature available today to actively measure and report many network metrics.
- It is easy to use, and is supported by many existing network management applications.
- We also have MPLS OAM, Gatekeeper Registration, H323/SIP Call Setup operation, and many other new features.
- Stay tuned… We have an ambitious roadmap for new features like better voice measurements and we're always listening your suggestions!

Recommended Reading

BRKNMS - 3004

- Practical Service Level Management
- Network Management: Accounting and Performance Strategies (Jul 07)
- Network Administrators Survival Guide

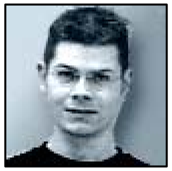


Available in the Cisco Company Store

Meet the Experts
Management & Operations

- Benoit Claise
  Distinguished Service Engineer
- Bruno Klauser
  Consulting Systems Engineer
- Emmanuel Tychon
  Technical Marketing Engineer
- Ralph Droms
  Technical Leader
- Stephen Mullaney
  Technical Marketing Engineer
- Stuart Parham
  Consulting Systems Engineer






Q and A